CONTINUOUS AREA ALLOCATION SYSTEM

Patent Number: JP6139703
Publication date: 1994-05-20
Inventor(s): MATSUMOTO SATOSHI; others: 02
Applicant(s): HITACHI LTD
Requested Patent: ☐ JP6139703
Application Number: JP19920288397 19921027
Priority Number(s):
IPC Classification: G11B20/12; G06F12/00
EC Classification:
Equivalents:

Abstract

PURPOSE:To improve access performance by successively bisecting all the areas of a RAM, arranging the leading blocks of respective files at dividing points, continuing blocks to this arranged block thereafter, finding out any place when the arranging place is occupied with the same method, and continuously arranging blocks to this place thereafter.
CONSTITUTION:1-1 to 1-7 shown in diagram are the leading blocks of files, and 4 is the RAM. The leading block 1-1 is arranged (2) at the position of a search pointer 2 from an initial state (1), and the pointer 2 is moved for 1/4. In the state (3) of generating two files, the pointer is moved for 1/8 so as to deal with the generation/erase of the file as prescribed. In a state (6), the pointer 2 shows 1/2 in a state (6) but since the block 1-1 is already arranged, the next 1/4 is investigated, the leading block 1-6 is arranged in the empty area, and the pointer is moved for 3/4. When the size of the file is small, continuous areas can be allocated to a lot of files. The files in a big size are allocated to several continuous areas. The number of times for the I/O of files is reduced.

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－139703

(43)公開日　平成6年(1994)5月20日

(51)Int.Cl.[5]	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｇ１１Ｂ　20/12		9295－5D		
Ｇ０６Ｆ　12/00	５０１　Ｈ	8526－5B		

審査請求　未請求　請求項の数３(全　６　頁)

(21)出願番号　特願平4－288397

(22)出願日　平成４年(1992)10月27日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　松本　智
横浜市戸塚区吉田町292番地株式会社日立製作所マイクロエレクトロニクス機器開発研究所内

(72)発明者　幡野　富久
横浜市戸塚区吉田町292番地株式会社日立製作所マイクロエレクトロニクス機器開発研究所内

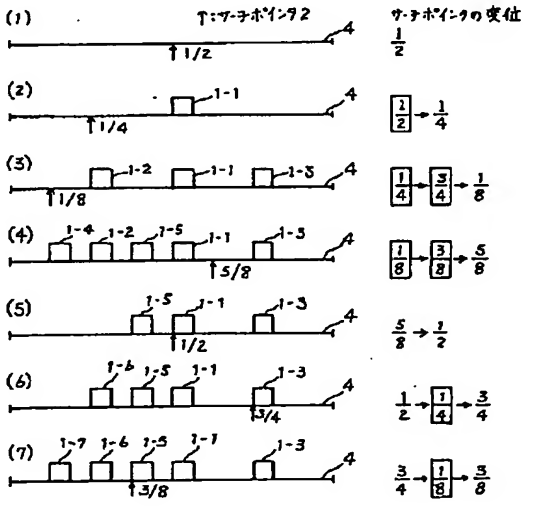
(74)代理人　弁理士　小川　勝男

最終頁に続く

(54)【発明の名称】　連続領域アロケーション方式

(57)【要約】
【目的】ランダムアクセス記憶装置内のファイルのフラグメンテーションを減らし、ファイルアクセス時のＩ／Ｏ回数を減らしてランダムアクセス記憶装置のアクセス性能を向上させること。
【構成】ランダムアクセス記憶装置を次々と二分割していき、その二分割点にファイルの先頭ブロックを割当て、ファイルの二番目以降のファイルは先頭ブロックの後方に連続して割り当てる。
【効果】隣合う二分割点の距離がファイルサイズよりも小さくなるまでファイルに連続領域を割り当てることができるため、ランダムアクセス記憶装置を効率良く使用しながらファイルアクセスを高速化できる。

図１



□は，ブロックがアロケーションされたことを示す。

1

【特許請求の範囲】
【請求項１】ランダムアクセス記憶装置において、前記記憶装置の全データ領域を次々と二分割していき、前記二分割点に各々のファイルの先頭ブロックを割り当てることを特徴とする連続領域アロケーション方式。
【請求項２】前記記憶装置において、前記二分割点に割り当てられた前記先頭ブロック以降の前記ファイル中のブロックを、前記先頭ブロックの直後に連続して割り当てることを特徴とする請求項１記載の連続領域アロケーション方式。
【請求項３】前記連続して割り当てたブロックの直後のブロックが既に割当て済みだった場合、前記ファイル中の次のブロックは前記先頭ブロックを割り当てる方法に従い割当て、それ以降の前記ファイル中のブロックは前記次のブロックの直後に連続して割り当てることを特徴とする請求項１記載の連続領域アロケーション方式。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、ランダムアクセス記憶装置にデータを格納するときの、ファイルの配置方式に関する。
【０００２】
【従来の技術】従来は図３に示すように、記憶装置を３－１，３－２，３－３，３－４・・・の固定サイズの連続領域に分割し、各ファイルを各連続領域に配置している。ファイル数が連続領域数を越えた場合、一つの連続領域中に複数のファイルが混在することもある。
【０００３】
【発明が解決しようとする課題】従来の技術では、ファイル数が連続領域数を越えると一つの連続領域に複数のファイルが混在し、連続領域を確保しきれなくなる。そのためＩ／Ｏ回数が増え、ファイルアクセスの性能が劣化する。
【０００４】
【課題を解決するための手段】上記課題を解決するために、ランダムアクセス記憶装置の全領域を次々に二分割していき、各々の二分割点に各ファイルの先頭ブロックを配置する。また、先頭ブロックの次以降のブロックは、先頭ブロックに連続するように配置する。さらに、連続するように配置しようとするブロックの配置場所が使用中の場合、そのブロックの配置場所は先頭ブロックの配置場所を見つけるのと同じ方法で見つける。それ以後のブロックはそのブロックに連続するように配置する。
【０００５】
【作用】ファイルが生成される度に連続領域が割り当てられるため、一つの連続領域に複数のファイルが混在することが無くなる。しかし、ファイル数が増えてくると連続領域の長さが短くなってくる。が、一つの連続領域に入りきらなかった部分は別の連続領域に配置されるため、それほどフラグメンテーションは激しくならない。

2

【０００６】
【実施例】本発明の一実施例を、図を用いて説明する。
【０００７】まず初めに、本発明の考え方について説明する。
【０００８】図１は本発明におけるファイルの先頭ブロックを配置していく過程を示した図である。１－１～１－７は各ファイルの先頭ブロックを示し、２はサーチポインタを示し、４はランダムアクセス記憶装置を示す。左側の図は記憶装置４中のどの部分にどのファイルの先頭ブロックが割り当てられたかを示し、右側の図はサーチポインタ２の変位を示している。
【０００９】（１）は初期化状態であり、記憶装置４上には一つもブロックが無く、サーチポインタは１／２の部分を示している。
【００１０】（２）は、（１）の初期化状態からファイルを一つ生成した状態を示す。ファイルの先頭ブロック１－１はサーチポインタ２が指し示していたところに配置され、その後、サーチポインタ２は１／４のところへ移動する。
【００１１】（３）は、（２）の状態からファイルを二つ生成した状態を示す。まず、一つ目のファイルの先頭ブロック１－２をサーチポインタ２が指し示している１／４のところに配置し、サーチポインタ２を３／４のところへ移動する。次に、二つ目のファイルの先頭ブロック１－３をサーチポインタ２が指し示している３／４のところに配置し、サーチポインタ２を１／８のところへ移動する。
【００１２】（４）は、（３）の状態からさらにファイルを二つ生成した状態を示す。まず、一つ目のファイルの先頭ブロック１－４をサーチポインタ２が指し示している１／８のところに配置し、サーチポインタ２を３／８のところへ移動する。次に、二つ目のファイルの先頭ブロック１－５をサーチポインタ２が指し示している３／８のところに配置し、サーチポインタ２を５／８のところへ移動する。
【００１３】（５）は、（４）の状態でファイルを二つ削除した状態を示す。まず、ブロック１－４を含むファイルを削除すると、サーチポインタ２は１／２にリセットされる。次に、ブロック１－２を含むファイルを削除すると、サーチポインタ２は再び１／２にリセットされる。
【００１４】（６）は、（５）の状態でファイルを一つ生成した状態を示す。サーチポインタ２は１／２の部分を指し示しているが、そこには既にブロック１－１が配置されているため、次の場所の１／４の部分をチェックする。そこはこの場合空き領域なので、そこにファイルの先頭ブロック１－６を配置し、サーチポインタ２を３／４に移動する。
【００１５】（７）は、（６）の状態でファイルを一つ

生成した状態を示す。サーチポインタ２は３／４の部分を指し示しているが、そこには既にブロック１－３が配置されているため、次の場所の１／８の部分をチェックする。そこはこの場合空き領域なので、そこにファイルの先頭ブロック１－７を配置し、サーチポインタ２を３／８に移動する。

【００１６】次に、ファイルの先頭のブロック以降のブロックの配置方法を図２を用いて説明する。

【００１７】１－１－１，１－１－２，１－１－３，１－１－４，１－１－５，１－１－６，１－１－７，１－１－８は、あるファイルを構成するブロックを示し、１－３－１，１－３－２は別のファイルを構成するブロックを示す。

【００１８】あるファイルの先頭ブロック１－１－１が図２に示す位置に配置されたとすると、それに引き続くブロック１－１－２，１－１－３，１－１－４は図２に示すように先頭ブロック１－１－１の直後に連続して配置される。その次のブロック１－１－５はブロック１－１－４の直後が塞がっているため、図１に示した方式により配置される。その結果、例えば図２に示す位置に配置される。それ以後のブロック１－１－６，１－１－７，１－１－８は、図２に示すようにブロック１－１－５の直後に連続して配置される。

【００１９】次に、本発明の具体的な実現方法について説明する。

【００２０】図４は、本発明を実現するにあたり必要となる情報テーブルの一例である。サーチポインタは図１の説明にもあるように、直前にアロケーションされたブロックの次にアロケーションされるべきブロックを指し示し、初期化時、及びファイル削除時に１／２にリセットされる。具体的には、図４に示すように分母５と分子６に分けて保持する。９はランダムアクセス記憶装置中に含まれる全ブロック数を示す。１０はランダムアクセス記憶装置中に含まれるブロックのうち、どのブロックが空きでどのブロックが使用中かを示す情報を保持するテーブルである。

【００２１】次に、ファイルの先頭ブロックを配置するアルゴリズムを示すＰＡＤを図４に示す。

【００２２】まず、７－１でサーチポインタの分母５が現在の値から全ブロック数９を越えるまで７－２～７－９を繰り返す。次に、７－２でサーチポインタの分子６が現在の値から現在のサーチポインタの分母５を越えるまで７－３～７－７を繰り返す。７－３では第（サーチポインタの分子６／サーチポインタの分母５×全ブロック数９）番目のブロックが空きかどうかを空きブロック管理テーブル１０を見て判定し、空きならば７－４～７－６を実行する。７－４ではサーチポインタの分子６に２を加える。７－５では、７－３で判定したブロックに対応する空きブロック管理テーブル１０のエントリを使用中にし、７－６では７－３で得たブロックの番号を返してリターンする。７－３で判定したブロックが空きでなかったとき、７－７を実行する。７－７ではサーチポインタの分子６を二つ増加する。７－２の判定条件が偽になったら７－８，７－９を実行する。７－８ではサーチポインタの分子６を１にし、７－９ではサーチポインタの分母５を二倍にしている。７－１の判定条件が偽になったら７－１０を実行する。７－１０では空きブロック管理テーブル１０中を先頭から一つずつ空き領域を探し、見つけた空きブロックに対応するエントリを使用中にし、見つけた空きブロックの番号を返す。

【００２３】次に、ファイルの二番目以降のブロックを配置するアルゴリズムを示すＰＡＤを図６に示す。

【００２４】まず、８－１でファイルの最後部のブロックの次のブロックに対応する、空きブロック管理テーブル１０のエントリをチェックする。８－２で、そのブロックが空きならば８－３，８－４を、そうでないならば８－５を実行する。８－３では８－１でチェックしたエントリを使用中にし、８－４では８－１でチェックしたエントリに対応するブロック番号を返してリターンする。８－５では図４で示した処理を行う。

【００２５】以上で本発明の一実施例の説明を終わる。

【００２６】

【発明の効果】ランダムアクセス記憶装置のデータ領域を次々と二分割してファイルに割り当てているため、ファイルの大きさが小さい場合、かなり多くのファイルに連続領域を割り当てることができる。また、サイズの大きいファイルもいくつかの連続領域に割り当てられるので、ファイルＩ／Ｏの回数を削減できる。記憶装置中のファイルの数が少ないとき、サイズの大きいファイルにも一つの連続領域を割り当てることができ、柔軟性にも富んでいる。

【図面の簡単な説明】

【図１】ファイルの先頭ブロックの配置方式を示した図である。

【図２】ファイルの二番目以降のブロックの配置方式を示した図である。

【図３】従来のファイルのブロックの配置方式を示した図である。

【図４】本発明を実現するために必要な情報を含むテーブルの一例を示す図である。

【図５】ファイルの先頭ブロックを配置するアルゴリズムを示したＰＡＤである。

【図６】ファイルの二番目以降のブロックを配置するアルゴリズムを示したＰＡＤである。

【符号の説明】

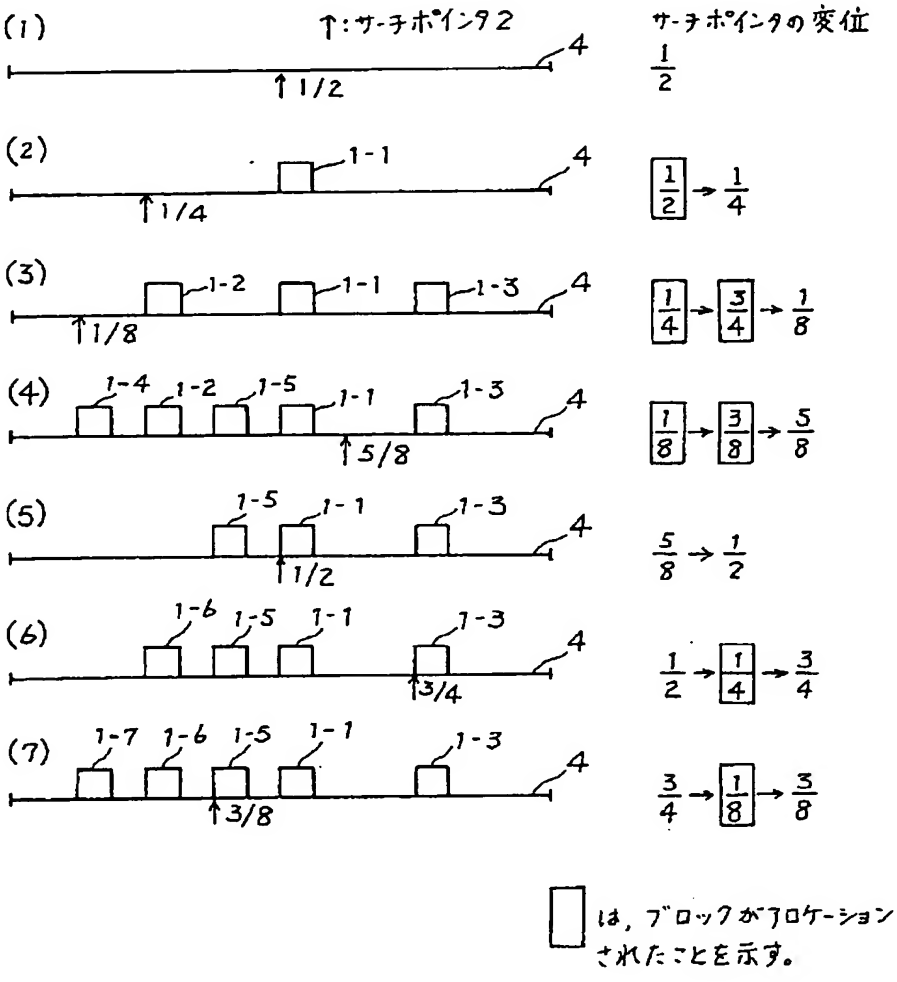
１－１～１－７…ファイルの先頭ブロック、２…サーチポインタ、３－１～３－４…連続領域、４…ランダムアクセス記憶装置のデータ領域、５…サーチポインタの分母、６…サーチポインタの分子、７－１～７－９…ファイルの先頭ブロックを配置するアルゴリズムを示したＰ

5　　6

ＡＤ、８－１～８－５…ファイルの二番目以降のブロックを配置するアルゴリズムを示したＰＡＤ、９…ランダムアクセス記憶装置中の全ブロック数、１０…ランダム＊アクセス記憶装置中のどのブロックが空きでどのブロックが使用中であるかを示す情報を含むテーブル。

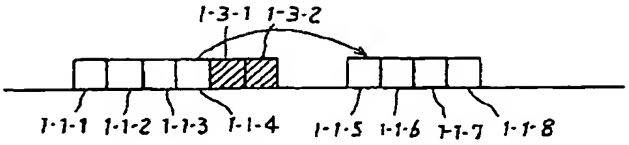
【図１】

図 1



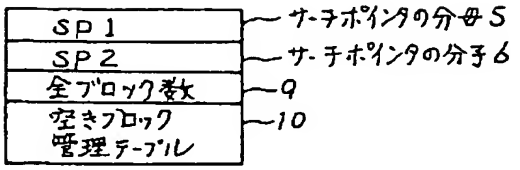
【図２】

図 2



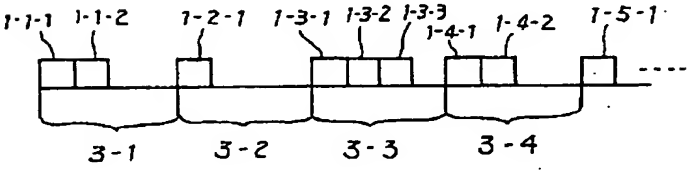
【図４】

図 4



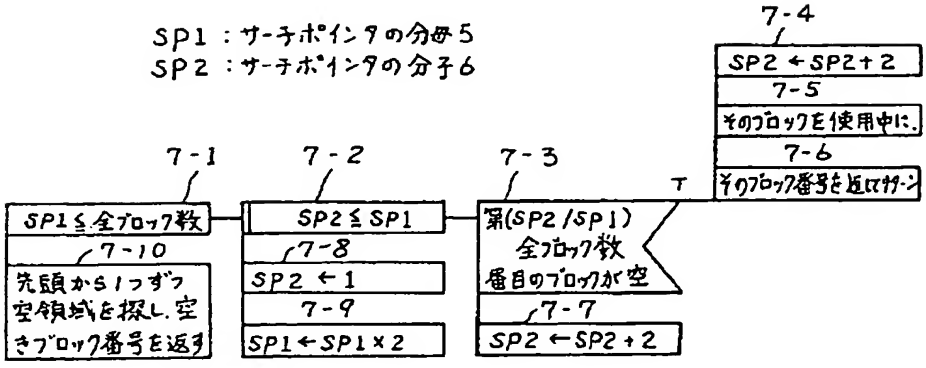
【図３】

図 3



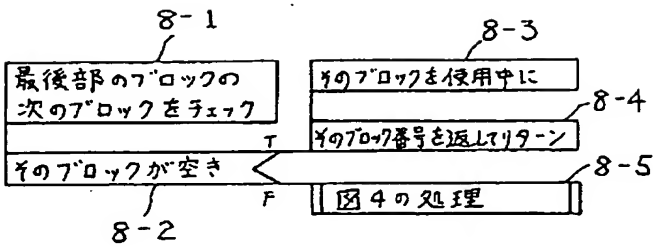
【図５】

図 5



【図６】

図 6



フロントページの続き

(72)発明者　岩田　吉隆
横浜市戸塚区戸塚町216番地株式会社日立
製作所情報通信事業部内